Research Topic: **Object Pooling**

Current game in development for selected research topic: **Tower Defence game**

Why you selected the research topic?

Tower defence is a game which requires a heavy flow on constant instantiation and destroying of units/minions. Also, a typical Tower defence game spawns around 100 units or minions in a given game time of around 10 minutes. This is a lot of gameobjects from a game programmer's point of view. Object Pooling solves this problems for most games which involve instantiation and destroying of multiple gameobjects and reusing them back. Some other games which use Object Pooling are First Person Shooter games (pooling bullets) , an average bullet hell game, RTS, MOBA and much more. Using pooling will make my tower defence much more efficient and I can constantly spawn huge amounts of minions/units in batches or together. Also, I am now able to run my game on low powered CPUs since instantiation and destroying is CPU intensive. Such tasks are responsible for CPU spikes and sometimes are also responsible for lag and freeze in games sometimes.

Knowledge of the topic prior to the research?

I had some info about the advantages and what object pooling was. But, I never had the time nor a test case or a prototype in which I could implement object pooling. This assignment gave me the opportunity to work on this I desired for so long. I had watches a lot of GDC videos in the past in which all pointed to have a sophisticated Object Pooling system.

Approach to researching the topic.

I first started watching YouTube videos having maximum video length of 10 minutes. This is because I could get a rough idea of what is object pooling. Once I understood what object is pooling, I first started to find out different ways in which this can be implemented. After that, I started to look at some Wikipedia pages and some indie blogs which have implemented object pooling in a different way. After reading and understand different approaches, I started reading about its advantages and disadvantages of object pooling and in what cases it is necessary to use it and when not to use it. After that, I found a good blog by catlikecoding.com regarding how to implement object pooling. The blog is very long and detailed. Although, I did not do exactly as the blog said, since I had to tailor the prototype as per my needs of the game and the assignment.

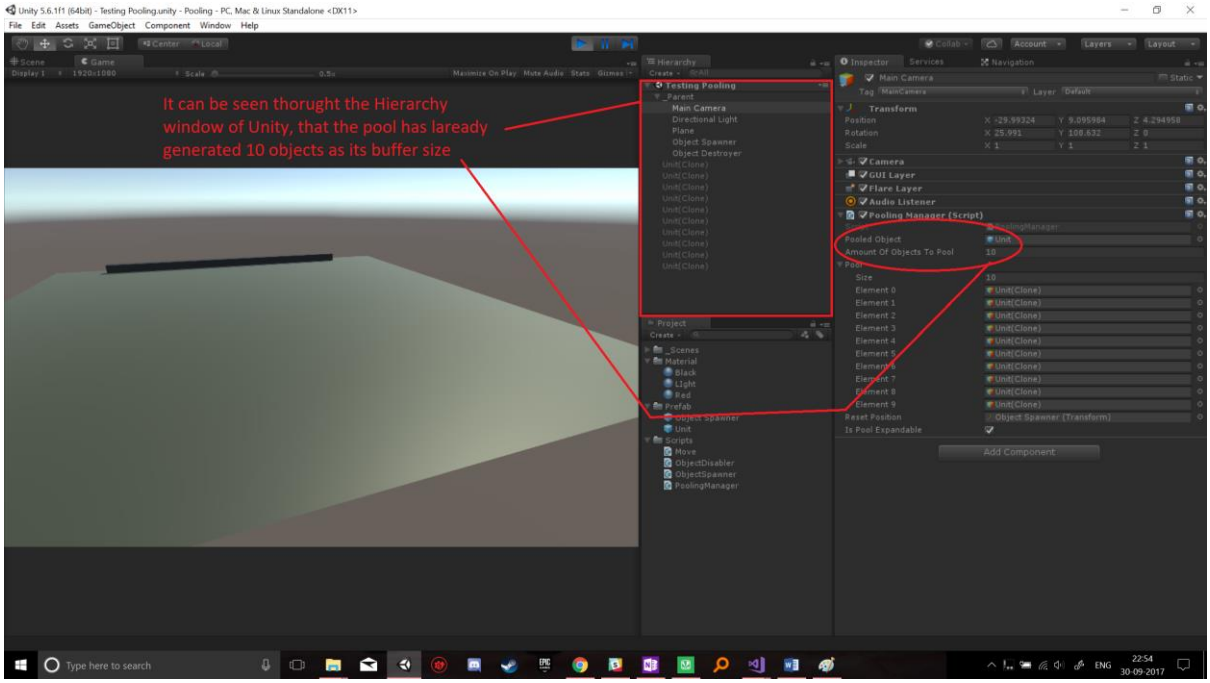
Learnt through you research.

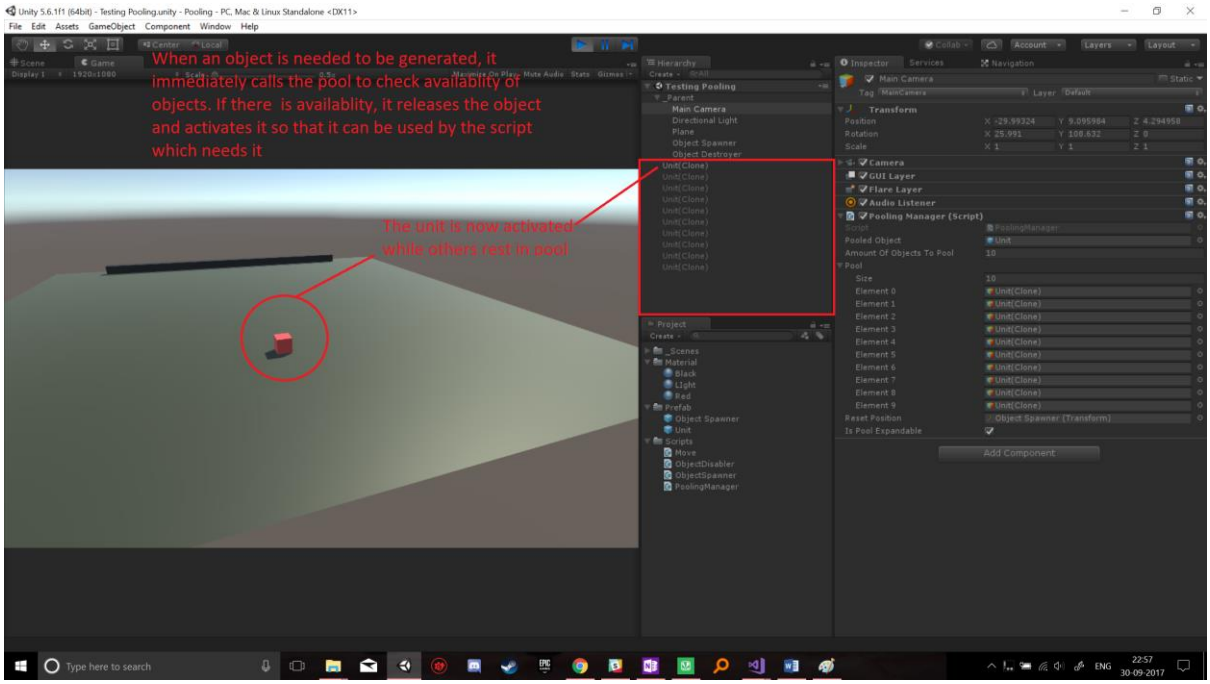
I know how to implement object pooling efficiently and use it in my game. I also learnt how to conserve and save precious CPU cycles on instantiation and destruction of objects and how costly memory is to a user.

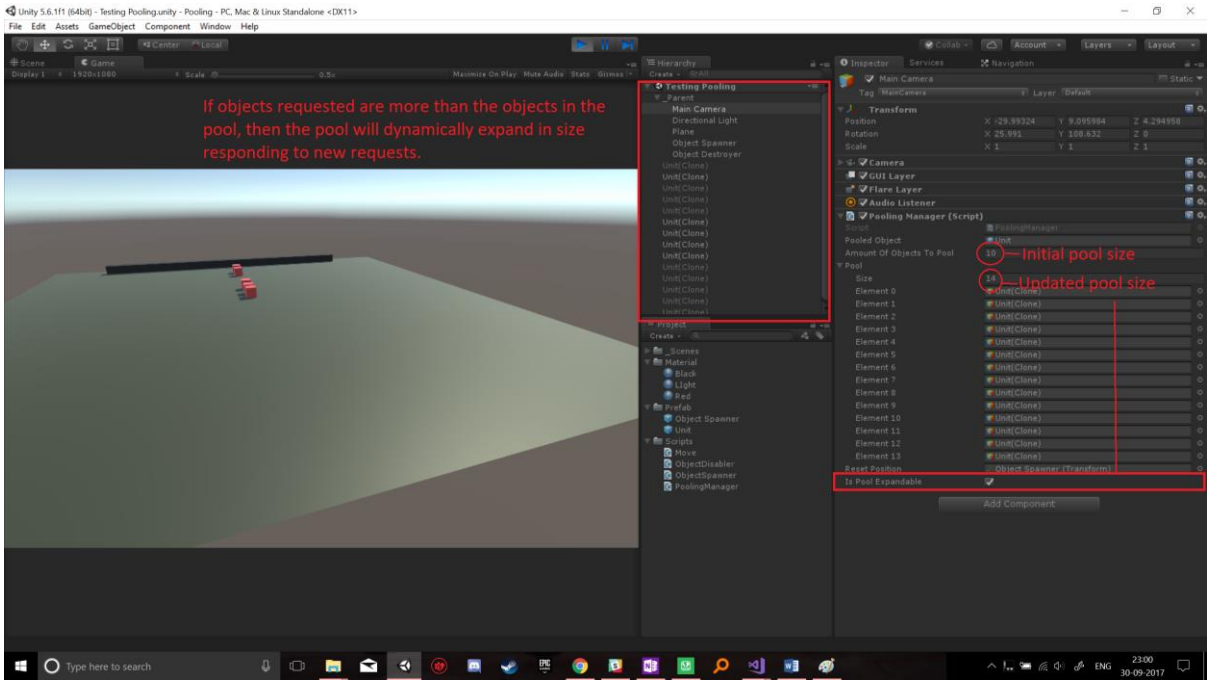
One other thing which the blog did not tell about was how to make the object pools dynamic. So I figured that out on my own and implemented it in the assignment. Now, the pools are dynamic and if the request comes in for more objects which is more than the current availability in pool, then the pool can handle such requests efficiently.

Some screenshots: (If there is difficulty in viewing the images just increase the zoom level on Microsoft word or adobe pdf reader.

The units are generated by pressing ‘space’.







Try the demo.

Open ‘Pooling’ folder and go to ‘_Scenes ‘and run it. Press ‘Space’ to spawn units.

Also included is the assignment description and the answers and a video to it.

Link to project files

https://github.com/aparant777/SimpleProjectLineups

Thanks,
